Uncertainty quantification in Discrete Fracture Network models: stochastic geometry

AGU PUBLICATIONS

Water Resources Research

RESEARCH ARTICLE
10.1002/2017WR021163

Uncertainty Quantification in Discrete Fracture Network Models: Stochastic Geometry

Stefano Berrone[1], Claudio Canuto[1], Sandra Pieraccini[2], and Stefano Scialò[1]

[1]Dipartimento di Scienze Matematiche, Politecnico di Torino, Turin, Italy, [2]Dipartimento di Ingegneria Meccanica e Aerospaziale, Politecnico di Torino, Turin, Italy

Key Points:
- Addressing uncertainty in geometry of fracture networks
- Flow simulations on stochastic discrete fracture networks
- Multilevel Monte Carlo method for uncertainty quantification in underground simulations

Correspondence to:
S. Pieraccini,
sandra.pieraccini@polito.it

Citation:
Berrone, S., Canuto, C., Pieraccini, S., & Scialò, S. (2018). Uncertainty quantification in discrete fracture network models: Stochastic geometry. *Water Resources Research*, *54*, 1338–1352. https://doi.org/10.1002/2017WR021163

Received 24 MAY 2017
Accepted 2 FEB 2018
Accepted article online 6 FEB 2018
Published online 27 FEB 2018

Abstract We consider the problem of uncertainty quantification analysis of the output of underground flow simulations. We consider in particular fractured media described via the discrete fracture network model; within this framework, we address the relevant case of networks in which the geometry of the fractures is described by stochastic parameters. In this context, due to a possible lack of smoothness in the quantity of interest with respect to the stochastic parameters, well assessed techniques such as stochastic collocation may fail in providing reliable estimates of first-order moments of the quantity of interest. In this paper, we overcome this issue by applying the Multilevel Monte Carlo method, using as underlying solver an extremely robust method.

1. Introduction

Several applications related to subsoil exploitation require to perform underground flow simulations, for example, in order to assess safety and viability of the procedure under investigation. Just to mention a few, examples include geothermal applications, aquifers monitoring, geological storage (e.g., CO_2 or nuclear waste storage). In many situations, the pattern of fractures in the underlying medium is relevant in order to establish directionality of the flow, and the medium is modeled by means of the so-called discrete fracture network (DFN) model. Within this framework, fractures in the rock matrix are modeled via a 3-D set of intersecting polygons. We consider here a single phase flow problem in an impervious rock matrix, so that the flow is confined in the DFN. Fractures are pathways for the flow, and flux exchange among fractures occurs through fracture intersections.

Since full deterministic data about the underground are not available, the fracture networks used for computations are stochastically generated starting from probabilistic distributions of geometrical and hydrogeological properties. The stochastic generation of DFNs is likely to generate geometrical configurations very challenging for the meshing process.

There is a vast literature on the simulation of flow in porous and fractured media, see for example (Ahmed et al., 2015; Brenner et al., 2016; Dershowitz & Fidelibus, 1999; Faille et al., 2016; Karimi-Fard et al., 2004; Lenti & Fidelibus, 2003; Martin et al., 2005; Sandve et al., 2012), to mention a few. Many works in the recent literature are especially targeted at tackling the geometrical complexity of DFNs, aiming at performing efficient and reliable simulations in complex networks. Among the others, in Pichot et al. (2010, 2012, 2014), the mortar method is used in order to allow for a partial nonconformity of the mesh at fracture intersections. Other strategies involve the use of nonstandard finite element techniques in order to ease the meshing process: the eXtended Finite Element Method (XFEM) has been used in Huang et al. (2011), D'Angelo and Scotti (2012), Berrone et al. (2013a), Formaggia et al. (2014), and Schwenck et al. (2015) allowing for meshes that can cross the interfaces; Mimetic Finite Differences (MFD) were used in Huang et al. (2014) and Al-Hinai et al. (2015); the Virtual Element Method (VEM) was applied in Benedetto et al. (2014, 2016a, 2016b), allowing for polygonal meshes. A new code for the simulation of flows in fracture networks, also taking into account randomness of input parameters is described in Hyman et al. (2014), Makedonska et al. (2015), and Hyman et al. (2015). Different approaches rely on dimensional reduction of the problem, as suggested in Nœtinger and Jarrige (2012) and Nœtinger (2015), or on modifications of the geometry to simplify the mesh generation process, as, e.g., in Mustapha and Mustapha (2007).

In several applications, a relevant quantity of interest is the overall flux flowing through the network. In this work, we focus on this quantity, nevertheless the results obtained can be extended to other quantities of

interest. Due to the large uncertainty in several parameters describing the subsoil, the problem of uncertainty quantification on the output of simulations is a relevant issue.

The main target of this paper is to present a viable technique for addressing this problem, in particular in the relevant case of uncertainty in the geometry of the network; this is especially challenging because randomness in the geometry of computational domain may yield a quantity of interest which is nonsmooth in the space of the stochastic parameters. This lack of smoothness is known to prevent a successful application of modern stochastic collocation strategies and as such calls for alternative approaches. We consider here a geometric Multilevel Monte Carlo (MLMC) method, introduced by Giles (2008), which improves the computational efficiency with respect to standard Monte Carlo method relying on several levels of accuracy in the space discretization, catching the stochastic behavior of the quantity of interest with possibly many samples at the less accurate (and cheaper) levels, while improving on the estimates with few samples at the finer levels. An effective application of this strategy requires that an underlying solver is available, which allows to use possibly very coarse meshes. In the framework of DFN simulations this is an issue, as the meshing process is known to be one of the major challenges. In very complex systems, for methods based on conforming meshes, the coarsest mesh compatible with geometrical constraints would count a relevant number of elements, therefore successive refinements would increase the number of unknowns yielding soon intractable problems. The solver proposed by Berrone et al. (2013a) is based on nonconforming meshes and has proven to be extremely effective in tackling complex geometrical configurations even with coarse meshes; this key property makes it a suitable tool to be used in conjunction with a multilevel strategy, allowing to use an affordable number of levels of refinement. The use of MLMC method in the framework of DFN simulations with uncertainty in the network geometry is therefore proposed here for the first time, relying on the robustness of the mentioned solver. Several numerical results show the reliability of this approach.

The paper is organized as follows. In section 2, we briefly sketch the flow model and recall the key features of the numerical method used to perform the simulations. In section 3, we discuss the stochastic description of the network and describe the method adopted for uncertainty quantification analysis. In section 4, we report some results highlighting the viability of the approach.

2. The Underlying Problem

Let us consider a discrete fracture network $\mathcal{D}$, given by the union of mutually intersecting polygonal fractures F_i, $i=1,\dots,I$. Fracture intersections are called traces, denoted by S_m, $m=1,\dots,M$. According to the DFN framework, traces are always segments, and, for simplicity, we assume that each trace is shared by exactly two fractures, such that there is a one-to-one relationship between each trace index m and a couple of fracture indices, denoted by i^m and j^m, being S_m the intersection of F_{i^m} and F_{j^m}. The hydraulic head H in $\mathcal{D}$ is governed by Darcy's law on each fracture, with additional matching conditions at fracture intersections, enforcing the continuity of H across the traces, and the volumetric balance of the trace-normal component of the flux of H at each trace. The boundary of $\mathcal{D}$ is given by the union of the boundaries of all the fractures F_i, $i=1,\dots,I$, and is exactly divided in a Dirichlet part Γ_D, on which a value of H is prescribed, and a Neumann boundary Γ_N, where a boundary-normal flux component is set. The hydraulic head H in $\mathcal{D}$ is then given by the solution of the system of partial differential equations on the fractures with the above mentioned interface conditions at the traces and boundary conditions.

This formulation of the problem is mathematically well posed, but its numerical solution can be cumbersome when an intricate network is considered. This is the case, for example, when standard techniques are applied which require conformity of the mesh at fracture intersections, namely, if the edges of mesh elements, on each fracture, are required to match the traces on the fracture (partial conformity) and the elements on the intersecting fractures (total conformity). Indeed, in these cases, fractures intersecting with extremely narrow angles can lead to elongated elements in the computational mesh, with a consequent loss of accuracy in the computed solution. Also the presence on a fracture of traces very close to each other, and/or traces with different length scales, can cause the introduction of a very large number of mesh elements in order to correctly reproduce all the geometrical features, if mesh conformity requirements have to be satisfied; in these cases, the element size is mainly determined by geometrical constraints rather than by the desired accuracy level.

Since the present work deals with the solution of the hydraulic head problem on networks with a random geometry, the use of a solver robust to the presence of the aforementioned geometrical complexities in the networks appears mandatory. In particular, the optimization-based solver proposed in Berrone et al. (2013a, 2013b, 2014a) was shown in Berrone et al. (2016) and Pieraccini and Scialò (2016) to be extremely robust against DFN complexities. In the aforementioned references, the hydraulic head problem in a DFN is reformulated as a PDE-constrained optimization problem, in which a cost functional, expressing the error in the imposition of the matching conditions at the traces, is minimized, constrained by the PDEs expressing Darcy's law on the fractures. The computation and the minimization of the functional can be performed without requiring any kind of mesh conformity at the traces; this key feature allows, in particular, to apply the method also with possibly very coarse meshes, independently of the complexity of the network. After discretization, the problem is written in terms of the discrete variables h_i, representing the discrete hydraulic head on fracture F_i, and u_{i^m}, representing the flux incoming fracture F_i through its trace S_m. In its simplest formulation, the discrete problem can be written as:

$$\min \sum_{m=1}^{M} ||h_{i^m} - h_{j^m}||^2 + ||u_{i^m} + u_{j^m}||^2 \quad \text{such that } Ah = Bu + f \tag{1}$$

where h_{i^m} denotes the restriction of h_i to S_m, f is a loading term, and matrices A and B arise from a finite element-like discretization of the PDEs independently written on all the fractures. We refer the reader to Berrone et al. (2014b) for a concise description of the method and to Berrone et al. (2013a) for full details. We remark that if the optimization problem is tackled via a gradient-based solver, the solution can be obtained by iteratively solving local problems on the fractures, the coupling arising from the minimization of the functional. Well-posedness of the local problems, which may be an issue for some approaches (see, e.g., Tartakovsky & Xiu, 2006; Xiu & Tartakovsky, 2006), is proven in Berrone et al. (2014a). The feature of splitting the overall problem in small local problems provides the method also with good parallel scalability performances (Berrone et al., 2015a).

3. Randomness Affecting the Network

Let $w=(h,u)$ be the exact solution of the mathematical model, which is assumed to depend on some stochastic parameters. Let $(\Omega, \mathscr{A}, \mathbb{P})$ be a probability space, being Ω the set of outcomes, $\mathscr{A}$ the σ-algebra of events, and $\mathbb{P} : \Omega \mapsto [0,1]$ the probability measure. For each outcome $\omega \in \Omega$, let $w(\omega)$ denote the corresponding solution and let $\Phi(\omega)$ denote the random variable representing the *quantity of interest* (QoI), which is typically a suitable function g of the solution, namely $\Phi(\omega)=g(w(\omega))$. In several relevant applications, the QoI is, e.g., the overall flux flowing through the network, or, in case of transport simulations, the concentration of a passive scalar in a selected point of the network.

The main target is in evaluating the first-order and second-order statistical moments of Φ, namely in computing mean value and variance of the quantity of interest:

$$\mathbb{E}[\Phi] = \int_\Omega \Phi(\omega) \mathrm{d}\mathbb{P}(\omega), \qquad \sigma^2[\Phi] = \int_\Omega (\Phi(\omega) - \mathbb{E}[\Phi])^2 \mathrm{d}\mathbb{P}(\omega).$$

In recent works Berrone et al. (2015b, 2017), the authors have addressed the issue of evaluating the impact of stochastic fracture transmissivities, assumed there to be uniform (i.e., constant in space) on each fracture of the network. The quantity of interest was in the earlier work the portion of the overall flux affecting a prescribed fracture, and in the later work, the concentration of a passive scalar at an outcrop in a time-dependent transport problem. According to the network size, the transmissivities were assumed to be independent stochastic variables with a known probability density function, or described by means of a stochastic field, approximated by a truncated Karhunen-Loeve expansion (Le Maître & Knio, 2010). In both cases, stochastic collocation strategies (Nobile et al., 2008; Xiu, 2010) have been used to estimate mean value and variance, proving to be very successful.

In the present framework, the geometry of the network is assumed to depend on random parameters. For the sake of simplicity, we consider here randomness only affecting fracture sizes; namely, we consider fractures with a rectangular shape, the two edge sizes being independent random variables. We remark that

other sources of randomness could be considered, namely, the position of the center of mass, the density, and the orientation.

Unlike the case considered in Berrone et al. (2015b), in which the authors prove that under proper assumptions the quantity of interest is a smooth function of the stochastic variables, in the present context the quantity of interest may display a nonsmooth behavior in the stochastic parameter space (see, e.g., Figure 5). This is due to the fact that while varying the fracture dimensions, the number of traces may significantly change, and this can suddenly change flux intensity and directionality.

This lack of smoothness, as well known, may prevent a successful application of stochastic collocation strategies for approximating the first-order and second-order moments. Possible alternative approaches can rely on the Monte Carlo method, which estimates the mean value by averaging samples of the random variable. The main drawback of Monte Carlo method is its computational cost, as it is indeed well known that achieving a root mean square error below a given accuracy ε requires a number of samples which is $\mathcal{O}(\varepsilon^{-2})$. Several improvements of the standard Monte Carlo method have been proposed in the recent literature, such as, for example, quasi-Monte Carlo method (Dick et al., 2013; Graham et al., 2011) or Multilevel Monte Carlo method (Cliffe et al., 2011; Giles, 2008, 2015), or a combination of the two (Kuo et al., 2017).

We resort here to the Multilevel Monte Carlo method (from now on labeled MLMC), whose main ideas are briefly sketched in the next subsection.

3.1. The Multilevel Monte Carlo Method

The main idea behind MLMC (Giles, 2008) is to sample the quantity of interest by using the numerical solution of the problem under investigation at several levels of accuracy; the first-order and second-order statistical moments are roughly approximated using (possibly many) samples obtained from the lower levels, while few samples at the higher levels are used to improve on the values of the moments. This way, the overall computational cost can be largely reduced with respect to the standard Monte Carlo method (from now on labeled sMC). In the framework of partial differential equations depending on stochastic parameters, the different levels can be based on different levels of grid refinement in the space discretization, the lowest and the highest level corresponding to the coarsest and the finest mesh, respectively; in this case, the method is labeled *geometric* MLMC. We limit ourself to the description of this case, which is the one adopted here. This approach has been successfully applied in the framework of PDEs with random coefficients in several recent papers, see, e.g., Cliffe et al. (2011) and Teckentrup et al. (2013); see also Icardi et al. (2016) for an application in the framework of pore-scale simulations and Lu et al. (2016) for an application to stochastic oil reservoir simulations. We remark that, in the framework of DFN simulations, the application of this approach strongly hinges upon the availability of an underlying solver capable to work with very coarse meshes.

In the following description, we closely follow Cliffe et al. (2011). Let us introduce a discretization parameter δ (e.g., a finite element mesh size), yielding N degrees of freedom; let w_N denote the corresponding numerical solution and $\Phi_N = g(w_N)$ the quantity of interest computed from w_N. For a fixed value of N, sMC builds the following estimators for the mean value $\mathbb{E}[\Phi_N]$ and variance $\sigma^2[\Phi_N]$, respectively:

$$\hat{\Phi}_N^{\mathrm{sMC}} := \frac{1}{M}\sum_{m=1}^{M} \Phi_N(\omega_m), \tag{2}$$

$$\hat{S}_N^{\mathrm{sMC}} := \frac{1}{M-1}\sum_{m=1}^{M} (\Phi_N(\omega_m) - \hat{\Phi}_N^{\mathrm{sMC}})^2, \tag{3}$$

where $\omega_m \in \Omega$, $m = 1, \ldots, M$ are independent samples. Note that these are unbiased estimators, i.e., $\mathbb{E}[\hat{\Phi}_N^{\mathrm{sMC}}] = \mathbb{E}[\Phi_N]$ and $\mathbb{E}[\hat{S}_N^{\mathrm{sMC}}] = \sigma^2[\Phi_N]$.

Assume that for $N \to \infty$ we have $|\mathbb{E}[\Phi_N - \Phi]| = \mathcal{O}(N^{-\alpha})$ for a positive constant α. A well-known result (see, e.g., Cliffe et al., 2011) states that the mean square error $\mathbb{E}[(\hat{\Phi}_N^{\mathrm{sMC}} - \mathbb{E}[\Phi])^2]$ is made of two contributions, which are related to the sampling error and to the discretization error, respectively. Namely, it is easily proven that

$$\mathbb{E}[(\hat{\Phi}_N^{\mathrm{sMC}}-\mathbb{E}[\Phi])^2]=\sigma^2[\hat{\Phi}_N^{\mathrm{sMC}}]+(\mathbb{E}[\hat{\Phi}_N^{\mathrm{sMC}}]-\mathbb{E}[\Phi])^2=\frac{1}{M}\sigma^2[\Phi_N]+(\mathbb{E}[\Phi_N-\Phi])^2. \tag{4}$$

Hence, the target of a root mean square error smaller than a given accuracy ε is reached by ensuring, e.g., both terms on the right-hand side of the previous equation being smaller than $\frac{1}{2}\varepsilon^2$, yielding the following requests on the number of samples and on the number of degrees of freedom, respectively:

$$M \gtrsim \varepsilon^{-2}, \qquad N \gtrsim \varepsilon^{-1/\alpha},$$

where the symbol $\gtrsim$ means that the inequality is satisfied up to a multiplicative constant independent of any parameter. Assuming that the cost of a single simulation is approximately $C_N \simeq N^\gamma$ for a given positive γ, one obtains an overall cost for sMC which is

$$C_{\mathrm{sMC}}=MC_N \sim \varepsilon^{-2-\gamma/\alpha}. \tag{5}$$

The key point in MLMC is that several levels of accuracy of the solution come into play: sampling at lower (and cheaper) levels is used to catch the stochastic behavior; the estimator is then improved by means of higher levels sampling. The optimal number of samples to be computed at each level is obtained by minimizing the variance of the estimator while maintaining the computational cost fixed.

In more detail, let us consider a family of space discretizations with mesh parameter δ_ℓ, $\ell=1,\dots,L$, yielding N_ℓ degrees of freedom, with $N_1 < N_2 < \cdots < N_L$. Let w_ℓ represent the numerical solution obtained with the ℓth level mesh, and let $\Phi_\ell=g(w_\ell)$ be the corresponding QoI. The starting point of MLMC is the telescopic sum

$$\Phi_L=\Phi_1+\sum_{\ell=2}^{L}(\Phi_\ell-\Phi_{\ell-1}).$$

For the sake of convenience of notation, we set

$$\Phi_0 := 0, \qquad Y_\ell := \Phi_\ell-\Phi_{\ell-1}.$$

Assume that $\Phi \simeq \Phi_L$. Linearity of expectation yields

$$\mathbb{E}[\Phi] \simeq \mathbb{E}[\Phi_L]=\sum_{\ell=1}^{L}\mathbb{E}[Y_\ell], \tag{6}$$

hence the quantity $\mathbb{E}[\Phi]$ can be estimated by $\mathbb{E}[Y_1]$ plus some corrections involving finer meshes. In order to estimate values $\mathbb{E}[Y_\ell]$, the one-level estimator (2) is applied, at each level $\ell=1,\dots,L$, based on M_ℓ samples, obtaining

$$\hat{Y}_\ell := \frac{1}{M_\ell}\sum_{m=1}^{M_\ell} Y_\ell(\omega_m). \tag{7}$$

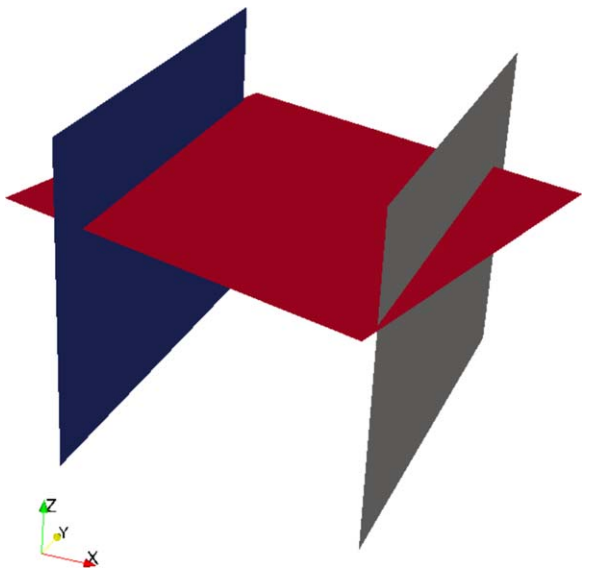

Figure 1. TP0. Geometry of the problem.

Note that one has

$$\mathbb{E}[\hat{Y}_\ell]=\mathbb{E}[Y_\ell], \qquad \sigma^2[\hat{Y}_\ell]=\frac{1}{M_\ell}\sigma^2[Y_\ell].$$

Concerning $\sigma^2[Y_\ell]$, note that this can be made arbitrarily small by increasing ℓ, thanks to the convergence in mean of Φ_N to Φ as $N \to \infty$.

Recalling (6), the multilevel estimator $\hat{\Phi}^{\mathrm{ML}}$ for $\mathbb{E}[\Phi]$ is built as

$$\hat{\Phi}^{\mathrm{ML}} := \sum_{\ell=1}^{L}\hat{Y}_\ell.$$

Since the Y_ℓ, for $\ell=1,\dots,L$, are independently estimated, standard formulas yield

$$\sigma^2[\hat{\Phi}^{\mathrm{ML}}]=\sum_{\ell=1}^{L}\frac{1}{M_\ell}\sigma^2[Y_\ell]. \tag{8}$$

Similarly to (4), the following relation for the mean square error is easily derived:

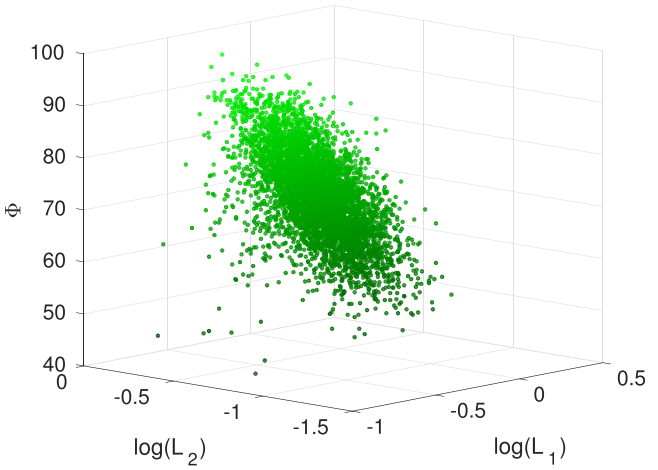

Figure 2. TP0. Flux measured versus values of the stochastic variables.

$$\mathbb{E}[(\hat{\Phi}^{\mathrm{ML}}-\mathbb{E}[\Phi])^2]=\sigma^2[\hat{\Phi}^{\mathrm{ML}}]+(\mathbb{E}[\hat{\Phi}^{\mathrm{ML}}]-\mathbb{E}[\Phi])^2=\sum_{\ell=1}^{L}\frac{1}{M_\ell}\sigma^2[Y_\ell]+(\mathbb{E}[\Phi_L-\Phi])^2.$$

As in the case of sMC, the target of reaching a root mean square error lower than a prescribed accuracy ε is attained by ensuring the two terms on the right-hand side being smaller than $\frac{1}{2}\varepsilon^2$. While the second one yields a limitation on N_L which is similar as in the sMC case, namely $N_L \gtrsim \varepsilon^{-1/\alpha}$, the first term can be made small at a cheaper computational cost than sMC, thanks to the fact that $\sigma^2[Y_\ell]$ decreases as ℓ increases; therefore, very few samples are likely to be needed at the higher levels, to guarantee a small overall variance of the estimator. In more detail, letting C_ℓ denote the computational cost of a single run at level ℓ, the overall computational cost for MLMC is

$$C_{\mathrm{ML}}=\sum_{\ell=1}^{L}C_\ell M_\ell. \tag{9}$$

The optimal M_ℓ, for each $\ell=1,\ldots,L$, is computed minimizing $\sigma^2[\hat{\Phi}^{\mathrm{ML}}]$ given by (8), treating M_ℓ as continuous variables, subject to the constraint of the overall cost C_{ML} defined in (9) being constant; the solution of this constrained minimization problem, yields at each level a value M_ℓ proportional to $\sqrt{\sigma^2[Y_\ell]/C_\ell}$, with a constant of proportionality depending on the accuracy ε^2 to be achieved.

The overall scheme for estimating the mean value is the following:

Multilevel Monte Carlo method

1. Compute $\hat{\Phi}_1$ as the one-level estimator (2) on a coarse mesh.
2. For each level $\ell > 1$, compute $\hat{Y}_\ell$ using (7) with the optimal number of samples M_ℓ.
3. Build the final estimator as:

$$\hat{\Phi}^{\mathrm{ML}}=\sum_{\ell=1}^{L}\hat{Y}_\ell.$$

The theoretical behavior of MLMC is summarized as follows (see Cliffe et al., 2011, Theorem 1; Giles, 2015, Theorem 1). Let us recall that we denote by N_ℓ the number of spatial degrees of freedom associated to the ℓth level mesh. Suppose that there exist constants $\alpha, \beta, \gamma > 0$, with $\alpha \geq \frac{1}{2}\min(\beta,\gamma)$, such that

$$|\mathbb{E}[\Phi_\ell-\Phi]|\lesssim N_\ell^{-\alpha}, \quad \sigma^2[Y_\ell]\lesssim N_\ell^{-\beta}, \quad C_\ell\lesssim N_\ell^{\gamma}.$$

Then, for any $\varepsilon < e^{-1}$, there exist L such that

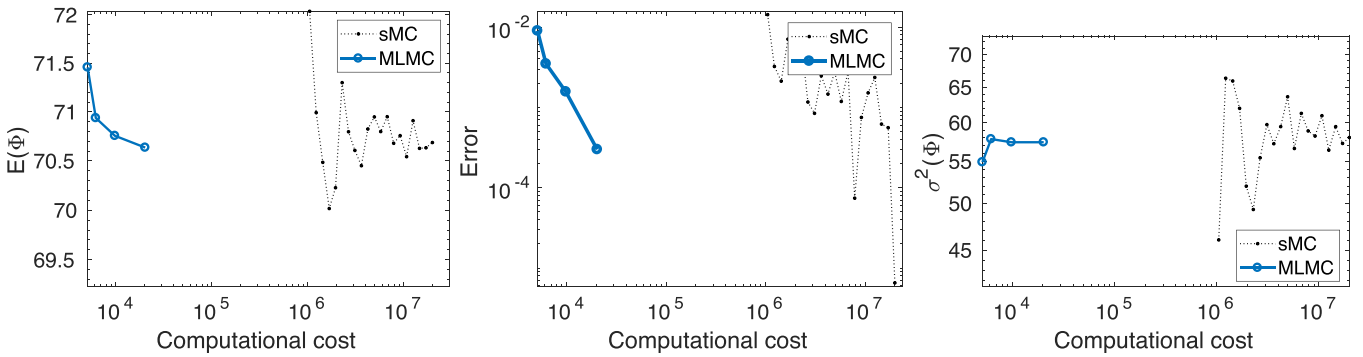

Figure 3. TP0. Results obtained with MLMC ($\varepsilon = 0.003$) compared to sMC. (left) Mean value, (middle) errors in the computation of the mean value, and (right) variance.

Table 1
Estimated Computational Costs (Floating Point Operations) for MLMC and sMC

	MLMC cost	sMC cost
TP0		
$\varepsilon = 0.002$	7.874e+04	1.675e+08
$\varepsilon = 0.003$	6.498e+04	7.443e+07
$\varepsilon = 0.006$	5.836e+04	1.861e+07
$\varepsilon = 0.009$	5.697e+04	8.270e+06
$\varepsilon = 0.012$	5.657e+04	4.652e+06
TP1		
$\varepsilon = 0.002$	1.825e+06	1.634e+09
$\varepsilon = 0.003$	1.823e+06	7.262e+08
$\varepsilon = 0.006$	5.550e+04	5.029e+06
$\varepsilon = 0.009$	5.533e+04	2.235e+06
$\varepsilon = 0.012$	5.528e+04	1.257e+06
TP2		
$\varepsilon = 0.002$	5.280e+04	1.176e+07
$\varepsilon = 0.003$	1.032e+04	5.226e+06
$\varepsilon = 0.006$	7.719e+03	1.306e+06
$\varepsilon = 0.009$	7.343e+03	5.806e+05
$\varepsilon = 0.012$	7.215e+03	3.266e+05

$$\mathbb{E}[(\hat{\Phi}^{\mathrm{ML}} - \mathbb{E}[\Phi])^2] < \varepsilon^2$$

with a total computational cost C_{ML} such that

$$C_{\mathrm{ML}} \lesssim \begin{cases} \varepsilon^{-2} & \text{if } \beta > \gamma \\ \varepsilon^{-2}(\log \varepsilon)^2 & \text{if } \beta = \gamma \\ \varepsilon^{-2-(\gamma-\beta)/\alpha} & \text{if } \beta < \gamma. \end{cases}$$

Few comments on this result are worthwhile. The result states that the behavior of MLMC, in terms of the total computational cost, is related to the relative values of β and γ, i.e., how fast the variance of $\Phi_\ell - \Phi_{\ell-1}$ gets reduced with respect to how fast the work load C_ℓ (for a single sample) increases with levels. Note that, due to previous comments, the overall cost at ℓth level is $C_\ell M_\ell \simeq \sqrt{\sigma^2[Y_\ell]C_\ell}$, and one has

$$C_{\mathrm{ML}} \simeq \sum_{\ell=1}^{L} \sqrt{\sigma^2[Y_\ell]C_\ell}.$$

If the case $\beta > \gamma$ applies, namely the variance gets reduced faster than the computational cost increases, the dominant term in C_{ML} is at the coarsest level; at this level, approximately $M_1 = \mathcal{O}(\varepsilon^{-2})$ samples have to be taken, but at a computational cost for each sample which is $\mathcal{O}(1)$; this compares well with the cost of sMC in which one would still take $M = \mathcal{O}(\varepsilon^{-2})$ samples but at a finer level. Recalling (5) we have a saving in MLMC which is $C_{\mathrm{ML}}/C_{\mathrm{sMC}} \simeq \varepsilon^{\gamma/\alpha}$. On the other hand, if one has $\beta < \gamma$, the dominant term in C_{ML} is at the finest level, in which $\mathcal{O}(1)$ samples are likely to be taken; the gain with respect to sMC is $C_{\mathrm{ML}}/C_{\mathrm{sMC}} \simeq \varepsilon^{\beta/\alpha}$. We remark that this latter case is the one occurring in our applications, see later in section 4.

Finally, as far as estimation of variance is concerned, following Bierig and Chernov (2015), the following multilevel estimator can be applied:

$$\hat{S}^{\mathrm{ML}} := \sum_{\ell=1}^{L} \left(\hat{S}_{M_\ell}[\Phi_\ell] - \hat{S}_{M_\ell}[\Phi_{\ell-1}]\right)$$

where for either $k = \ell$ or $k = \ell - 1$, we have set

Table 2
Number of Samples Used at Each Level

	$\varepsilon = 0.002$	$\varepsilon = 0.003$	$\varepsilon = 0.006$	$\varepsilon = 0.009$	$\varepsilon = 0.012$
TP0					
$\ell = 1$	12,716	5,965	1,452	644	327
$\ell = 2$	149	75	22	7	5
$\ell = 3$	8	5	5	5	5
$\ell = 4$	1	1	1	1	1
TP1					
$\ell = 1$	11,335	8,025	265	102	55
$\ell = 2$	122	135	5	5	5
$\ell = 3$	11	12	5	5	5
$\ell = 4$	1	1	1	1	1
$\ell = 5$	1	1	0	0	0
TP2					
$\ell = 1$	9,447	2,598	622	266	145
$\ell = 2$	316	33	5	5	5
$\ell = 3$	32	5	5	5	5
$\ell = 4$	5	1	1	1	1

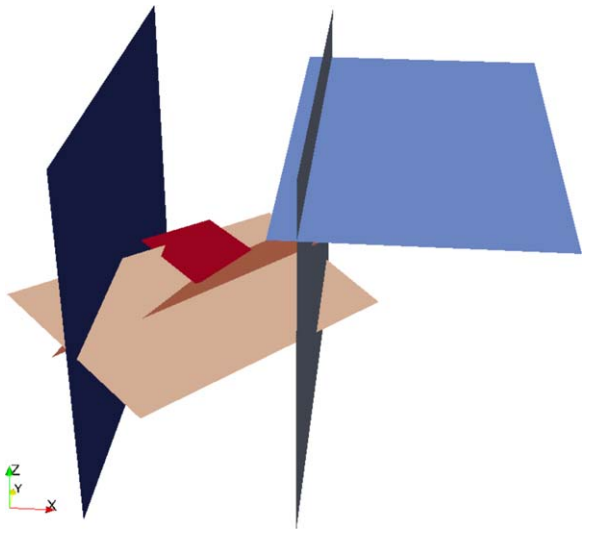

Figure 4. TP1. Geometry of the problem.

$$\hat{S}_{M_\ell}[\Phi_k] := \frac{1}{M_\ell - 1}\sum_{m=1}^{M_\ell}\left(\Phi_k(\omega_m) - \hat{\Phi}_k^{(\ell)}\right)^2, \qquad \Phi_k^{(\ell)} = \frac{1}{M_\ell}\sum_{m=1}^{M_\ell}\Phi_k(\omega_m).$$

4. Numerical Results

In this section, we show the effectiveness of MLMC in computing the mean value and variance of the QoI in the context of DFN flow simulations. We consider four DFN test problems with different characteristics, while sharing a common structure. In the first three test problems, two vertical fractures are present, representing a source fracture and a sink fracture. This is realized prescribing a value of the hydraulic head H on the top edge of such fractures ($H = 100$ and $H = 0$, respectively). In the fourth test problem, an inlet plane and an outlet plane are considered: a value $H = 10$ of the hydraulic head is prescribed on the intersections of fractures with the inlet plane, and $H = 0$ is set on intersections of fractures with the outlet plane. In all the test problems, all other fracture edges are treated as insulated. The quantity of interest is the overall flux flowing through the network, from the source to the sink fracture.

In each test case, MLMC has been applied in order to estimate the mean value and variance of the flux, normalized with respect to the maximum value attained in all the simulations, in such a way that the actual quantity of interest ranges in the interval $[0, 1]$ and the ε values can be considered as relative accuracies. All the figures in this section report the actual values, without normalization. We have applied a geometric MLMC, in which each level is characterized by a different spatial mesh, with an increasing number of dofs as the level number increases, so that level $\ell = 1$ corresponds to the coarsest mesh, which is a triangular mesh characterized by a minimum number of elements on each fracture equal to 25. In the first three test problems, at each new level, meshes are refined by splitting each triangle at the previous level into four smaller similar triangles, and then adding degrees of freedom corresponding to XFEM enrichments, which is the space discretization technique adopted herein (see Berrone et al., 2013b, 2016). In the fourth problem, standard FEM is used, in order to reduce the computational cost; at each level the number of triangles required is again multiplied by a factor 4.

As far as the work load is concerned, the main computational effort is related to the solution of the linear systems yielding the numerical solution. The first three test problems here proposed are small enough that a direct solver can be applied, and the cost is essentially related to the LDL^T factorization of the symmetric, indefinite matrix of the linear system providing the solution of the constrained minimization problem (1). In the fourth test, due to the large scale of the problems, the linear system is iteratively solved; in this case, the computational cost is related not only to the number of dofs, which determine the cost of a single iteration,

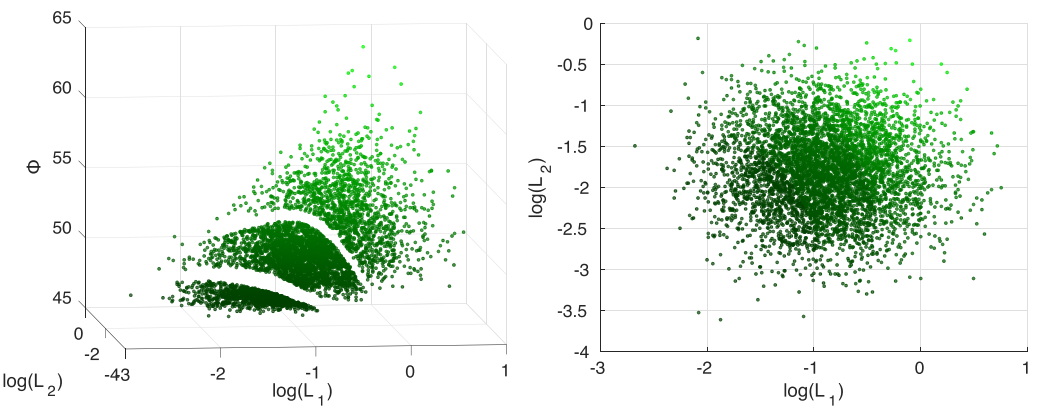

Figure 5. TP1. Flux measured versus values of the stochastic variables (right: a top view).

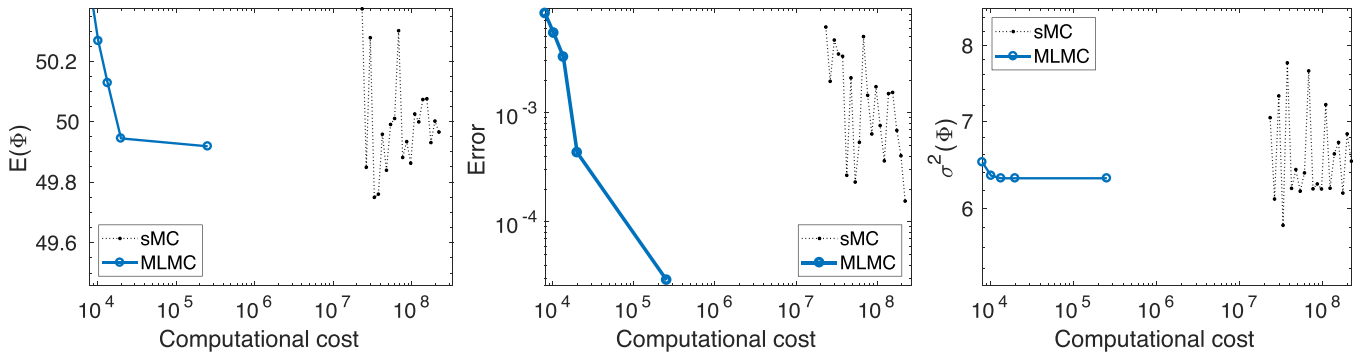

Figure 6. TP1. Results obtained with MLMC ($\varepsilon = 0.003$) compared to sMC. (left) Mean value, (middle) errors in the computation of the mean value, and (right) variance.

but also on the number of iterations performed, which clearly depends on the stopping criterion. We adopt here an *inexact* framework, namely, the linear systems are solved at an accuracy which increases with levels, using a stopping criterion with a relative tolerance which decreases with ℓ.

4.1. Test Problem TP0

The first test problem is a very simple problem, aiming at showing the accuracy attainable with the approach adopted herein with network configurations that are not yielding troublesome situations. The network, depicted in Figure 1, is made of two vertical deterministic fractures (the source and the sink), and a horizontal fracture with stochastic dimensions (the red one, in the figure). The two edge sizes of the horizontal fracture are assumed to be two independent stochastic variables L_1 and L_2 with log-normal distribution, namely, $\log(L_1) \sim \mathcal{N}(\mu_1, \sigma_1)$ and $\log(L_2) \sim \mathcal{N}(\mu_2, \sigma_2)$, being μ_1, μ_2 the average values and σ_1, σ_2 the standard deviations. More in details, we have taken

$$\mu_1 = -0.2007,\ \mu_2 = -0.5573, \qquad \sigma_1 = \sigma_2 = 0.15.$$

Since, in this case, the variability of fracture sizes does not yield fracture detachments, the overall flux crossing the network is a smooth function of the stochastic parameters. The QoI versus the images of the stochastic variables is depicted in Figure 2, in which we report the values of the computed flux versus the sampled values of $\log(L_1)$ and $\log(L_2)$. The number of dofs ranges from approximately 300 (level $\ell = 1$) to approximately 8,500 ($\ell = 4$). Different MLMC simulations with several values of the accuracy ε, namely $\varepsilon = 0.003$, 0.006, 0.009, 0.012, have been performed. With reference to the theoretical result on the behavior of MLMC reported in section 3.1, the values of the parameters α, β, and γ have been numerically estimated using linear regression and assuming that, for each ℓ, one has $N_\ell = sN_{\ell-1}$ for a suitable factor s; the values obtained for this test problem are approximately $\alpha = 0.7$, $\beta = 1.4$, and $\gamma = 3.1$, with an average growth factor $s = 3.2$.

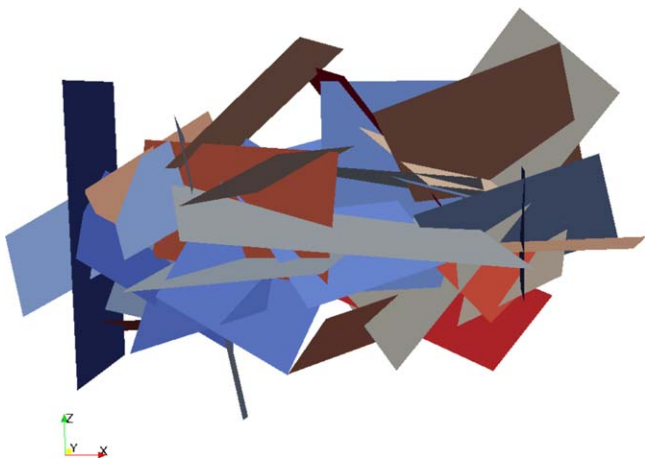

Figure 7. TP2. Geometry of the problem.

In Figure 3, left, we report, for $\varepsilon = 0.003$, the estimated value of $\mathbb{E}[\Phi]$ versus the computational cost. Similar behaviors are obtained for the other values of ε. The unit of measure for the x axis is the cost of one simulation on the coarsest level used for MLMC. The computational cost is measured in terms of estimated floating point operations, considering that the major source of computational effort for each run is the solution of the linear system yielding the numerical solution. In the same figure, we also report the value of the estimator computed with sMC on the finest mesh used for MLMC, which corresponds to $L = 4$. Note that the number of levels to be activated is automatically selected by the method in order to guarantee $\mathbb{E}[(\hat{\Phi}^{\mathrm{ML}} - \mathbb{E}[\Phi])^2] < \varepsilon^2$, see Giles, 2015. We observe that MLMC behaves quite well, compared to sMC; in particular, it remarkably computes the final estimator at a computational cost which is several orders of magnitude smaller than

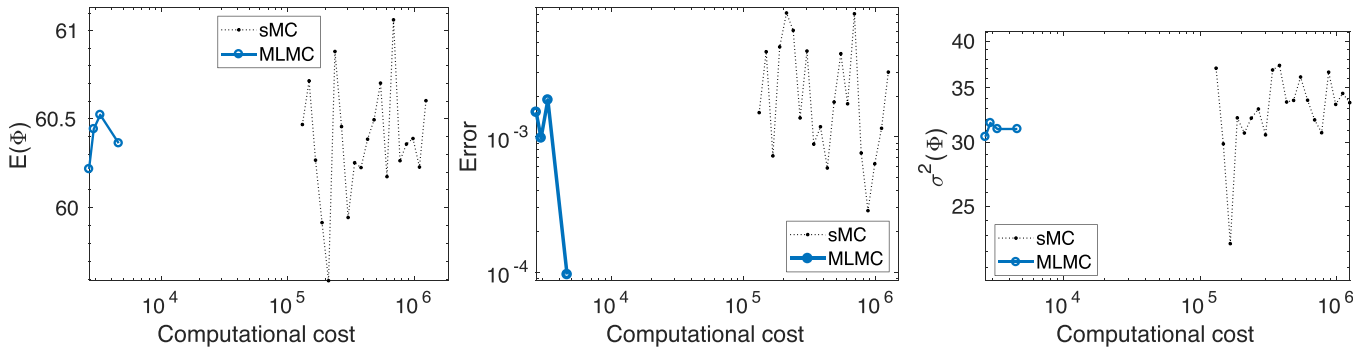

Figure 8. TP2. Results obtained with MLMC ($\varepsilon = 0.003$) compared to sMC. (left) Mean value, (middle) errors in the computation of the mean value, and (right) variance.

the one required by sMC; note also that the estimator computed by the latter method has still an erratic behavior.

In Figure 3, middle, we report for the same value of ε as the left figure, the relative error in the estimation of the mean value versus the computational cost, for both MLMC and sMC. The error is computed with respect to a reference solution obtained with a smaller ε for MLMC ($\varepsilon = 0.002$) and with a higher number of samples for sMC. It can be seen that the error gets reduced remarkably fast with MLMC, and a fixed error is reached by MLMC at a computational cost which is much lower than sMC. Finally, in Figure 3, right, we report the variance computed by both methods. Again, MLMC outperforms sMC in estimating the variance.

We can gain more insight into the computational cost of MLMC method by looking at Tables 1 and 2. Table 1 reports the theoretical costs predicted for MLMC and sMC methods versus some selected accuracies ε. It can be seen that, for all considered values of ε, MLMC is clearly expected to outperform sMC in attaining a prescribed accuracy at an affordable computational cost, which is typically 2–3 orders of magnitude smaller than sMC method. Table 2 reports the number of samples M_ℓ used at each level for the same accuracies ε used in the previous table. We may note that the number is rather high at the lowest level, but it rapidly decreases as the level increases; for example, with the smaller considered ε, we have $M_1 \simeq 10^4$, which quickly drops to $M_2 \simeq 10^2$, $M_3 \simeq 10$, and $M_4 = 1$. Most of the samples are therefore taken at the very coarse level (we recall that $\ell = 1$ corresponds to a few dozen triangles on each fracture), whereas at the fine levels, those at which one normally would compute the samples for sMC, just few samples are taken.

4.2. Test Problem TP1

This problem, while having a very simple geometry, yields a quite challenging behavior, which is typical of networks with stochastic geometry. The geometry is depicted in Figure 4. In this problem, five fractures

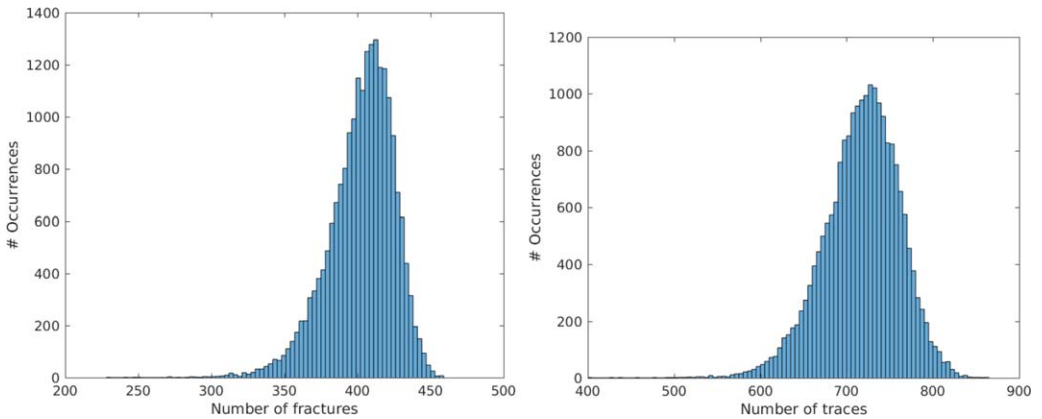

Figure 9. TP3. Distribution of the number of (left) fractures and (right) traces among the simulations.

Figure 10. TP3. Geometry of the problem.

have a deterministic size, whereas a sixth fracture has stochastic dimensions (the red one, in Figure 4). The number of dofs ranges from approximately 1,000 (level $\ell=1$) to approximately 70,000 ($\ell = 5$).

Also, in this case, the two edges of the nondeterministic fracture are assumed to be two independent stochastic variables L_1 and L_2 with log-normal distribution defined by the parameters

$$\mu_1 = -0.8863,\ \mu_2 = -1.8026, \qquad \sigma_1 = \sigma_2 = 0.5.$$

As the dimensions of this nondeterministic fracture change, it progressively detaches from the other fractures. Consequently, the number of traces ranges from 10, when the stochastic fracture does intersect all but one fracture, down to 6, when the stochastic fracture intersects no other fracture in the network, thus becoming excluded from the flux distribution along the network. These detachments cause a sudden change in the flux intensity; thus, the function describing the flux in the stochastic parameter space displays a nonsmooth behavior with several irregularity interfaces. This is shown in Figure 5 which, similarly to Figure 2, reports the computed fluxes versus the sampled values of $\log(L_1)$ and $\log(L_2)$. The right part of the figure shows a top view; since the colorbar is proportional to the value of the flux, the two steep flux variations that can be seen in the left plot correspond, in the top view, to the sudden change in color.

The MLMC method was applied to this problem with the same ε values used in the previous test case. Here the values estimated for the parameters α, β, and γ are approximately $\alpha=0.6$, $\beta=0.7$, and $\gamma=3.2$, with an average growth factor $s = 3.0$. The behavior is summarized in Figure 6, in which the same quantities as in Figure 3 are shown for this example. In particular, the middle figure reports the errors computed with respect to a reference solution obtained with $\varepsilon = 0.002$ (MLMC) and with a higher number of samples (sMC). It can be noted that, despite the quantity of interest exhibits a nonsmooth behavior, MLMC behaves quite well. Remarkably, the results obtained are quite similar to those attained in the smooth test case TP0,

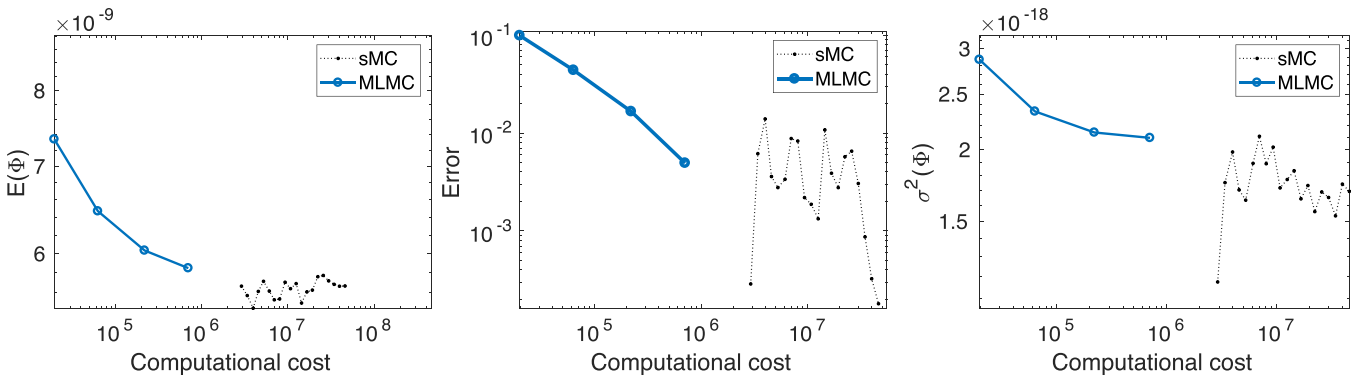

Figure 11. TP3, case a. Results obtained with MLMC ($\varepsilon = 0.003$) compared to sMC. (left) Mean value, (middle) errors in the computation of the mean value, and (right) variance.

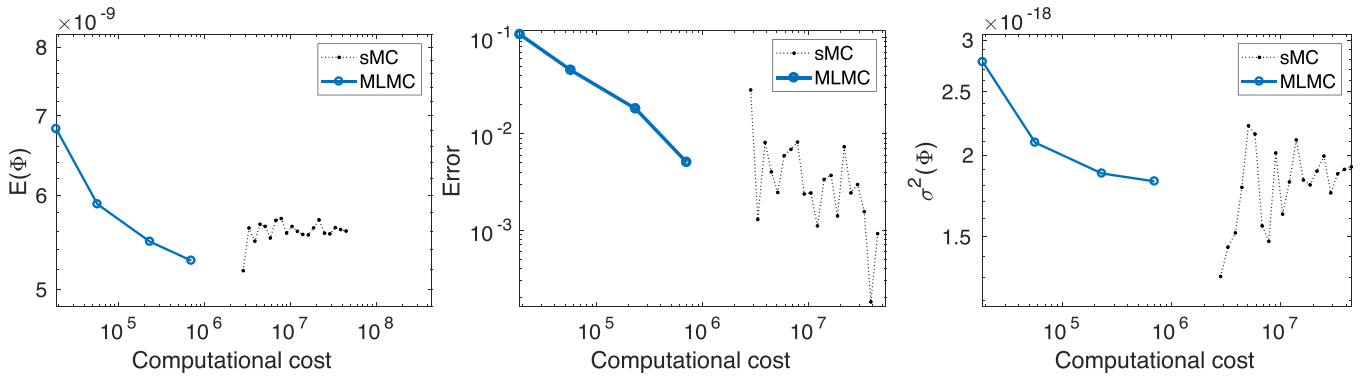

Figure 12. TP3, case b. Results obtained with MLMC ($\varepsilon = 0.003$) compared to sMC. (left) Mean value, (middle) errors in the computation of the mean value, and (right) variance.

meaning that the possible presence of nonsmoothness in the parameter space does not significantly affect the behavior of MLMC. The same conclusions can be driven by the analysis of Tables 1 and 2.

4.3. Test Problem TP2

The geometry of the third problem is depicted in Figure 7. In this case, we still consider two vertical fractures acting as a source and a sink (the dark blue ones in the figure), but an overall number of 31 fractures is present in the network. Among all these fractures, 8 are assumed to be deterministic (including the vertical ones), whereas the remaining 23 are stochastically generated. Each fracture in this latter group is assumed to have, as in the previous test problems, edge sizes modeled as two independent stochastic variables with log-normal distribution. The stochastic dimension is therefore $d = 46$. In this test problem, the connectivity changes within simulations as the number of traces ranges from 86 to 150, and also in this case a nonsmooth behavior is expected. The number of dofs ranges from approximately 10,000 (level ℓ=1) to approximately 125,000 ($\ell = 4$). The behavior of MLMC is again summarized in Figure 8 and Tables 1 and 2. The errors reported in the middle part of Figure 8 are obtained as in the previous cases. Here the values estimated for the parameters α, β, and γ are α=0.8, β=1.2, and γ=3.2, with an average growth factor $s = 2.4$. Also, in this case, MLMC is remarkably successful in providing a reliable estimate of the first-order and second-order moments at an affordable computational cost.

4.4. Test Problem TP3

The last problem is a more realistic one. Here we consider two vertical planes representing an inlet plane and an outlet plane, according to the imposed boundary conditions; an overall number of 520 fractures is stochastically generated in between, with edge sizes modeled as independent stochastic variables with log-normal distribution. The connectivity of the network is then analyzed, and only the maximal group of connected fractures is kept for the simulations. The total number of connected fractures present in each

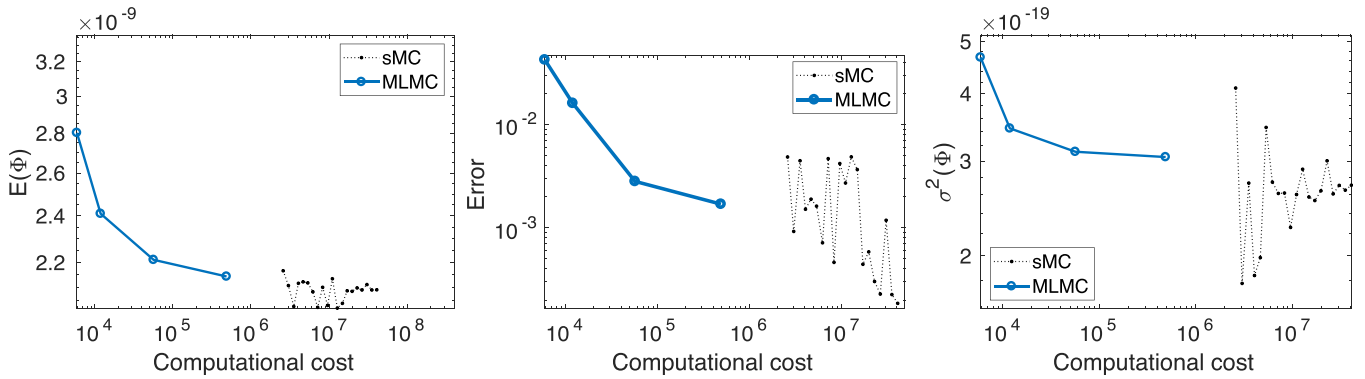

Figure 13. TP3, case c. Results obtained with MLMC (ε=0.005) compared to sMC. (left) Mean value, (middle) errors in the computation of the mean value, and (right) variance.

simulation is therefore smaller than 520. In Figure 9, we report the empirical distributions of the actual number of fractures and traces occurring in approximately 20,000 simulations considered. An example of network geometry is depicted in Figure 10, in which we also report, using a color scale, the numerical hydraulic head computed; in the figure, the left part corresponds to the inlet region, whereas the right part corresponds to the outlet region. The number of dofs ranges from approximately 15,000 to approximately 30,000 at the coarsest level ($\ell=1$), and from approximately 3×10^6 to approximately 4.5×10^6 at the finest level ($\ell = 5$).

In this last test problem, we also analyze the effects of a random transmissivity, in addition to randomness in fracture sizes. To this aim, we consider three different models for the fracture transmissivities:

Case a: The transmissivity is constant on all fractures, and is set to $K=10^{-9}$.
Case b: An *uncorrelated* model is adopted (see, e.g., Hyman et al., 2016): the transmissivity is considered a random variable with a log-normal distribution, namely $\log K \sim \mathcal{N}(\mu,\sigma)$, with $\mu=\log(10^{-9})$ and $\sigma=0.7$ (values are loosely based on data available in Svensk Kärnbränslehantering (2010)).
Case c: A *semicorrelated model* is considered, in which transmissivity is related to fracture size, with an additional stochastic term: namely, we consider on each fracture $\log K=\log(cA^d)+y$, where A is the fracture area and $y \sim \mathcal{N}(0,\sigma)$; the parameters used, loosely based on those reported in Svensk Kärnbränslehantering (2010), are $c=10^{-9}$, $d = 0.25$, and $\sigma=0.7$.

The values estimated for the parameters α, β, γ are: $\alpha=0.7$, $\beta=1.3$, and $\gamma=3.1$ with average growth factor $s = 3.5$ (case a); $\alpha=0.8$, $\beta=1.4$, and $\gamma=3.2$ with average growth factor $s = 3.3$ (case b); and $\alpha=0.7$, $\beta=1.3$, and $\gamma=3.2$ with average growth factor $s = 3.3$ (case c). Figures 11–13 report the results obtained in the three cases with $\varepsilon = 0.003$ (cases a and b) and $\varepsilon=0.005$ (case c); we used in this latter case a higher ε value as the computations turned out to be more expensive on this problem. Remarkably, the behavior of MLMC is quite similar in the three cases, and also quite similar to the previous cases. The approach seems therefore to be quite robust with respect to the network size and to the possible presence of several stochastic parameters with different natures.

5. Conclusion and Perspectives

In the framework of underground flow simulations in discrete fracture networks, we have proposed a viable approach to address the problem of uncertainty quantification on the output of simulations. In particular, we have dealt with the challenging and relevant case of networks with uncertainty in fracture sizes. The approach is based on the combination of the Multilevel Monte Carlo method with an extremely robust numerical solver able to deal with any kind of geometrical complexity which may arise. The results are remarkably promising, also when the quantity of interest is a nonsmooth function of the stochastic parameters. Future work will be devoted to analyze the behavior of the approach when other nondeterministic geometrical parameters are taken into account, such as position of center of mass and orientation. Furthermore, other strategies which may improve the method will be considered, such as, e.g., Multilevel quasi-Monte Carlo methods.

Acknowledgments
The authors are members of the INdAM research group GNCS. This research has been partially supported by the INdAM-GNCS. Computational resources were partially provided by HPC@POLITO computational facility of Politecnico di Torino (http://hpc.polito.it) and by CINECA trough Project "UQ-DFN" IsC45-HP10CJB18R. Most of the data used are reported in the text. Additional data which may be needed to reproduce the reported results are available from the corresponding author upon request.

References

Ahmed, R., Edwards, M., Lamine, S., Huisman, B., & Pal, M. (2015). Control-volume distributed multi-point flux approximation coupled with a lower-dimensional fracture model. *Journal of Computational Physics*, *284*, 462–489.

Al-Hinai, O., Srinivasan, S., & Wheeler, M. F. (2015). Domain decomposition for flow in porous media with fractures. In *SPE Reservoir Simulation Symposium 23–25 February 2013*. Houston, TX: Society of Petroleum Engineers.

Benedetto, M., Berrone, S., Borio, A., Pieraccini, S., & Scialò, S. (2016b). A hybrid mortar virtual element method for discrete fracture network simulations. *Journal of Computational Physics*, *306*, 148–166.

Benedetto, M., Berrone, S., Pieraccini, S., & Scialò, S. (2014). The virtual element method for discrete fracture network simulations. *Computer Methods in Applied Mechanics and Engineering*, *280*, 135–156.

Benedetto, M., Berrone, S., & Scialò, S. (2016a). A globally conforming method for solving flow in discrete fracture networks using the virtual element method. *Finite Elements in Analysis and Design*, *109*, 23–36.

Berrone, S., Canuto, C., Pieraccini, S., & Scialò, S. (2015b). Uncertainty quantification in discrete fracture network models: Stochastic fracture transmissivity. *Computers & Mathematics with Applications*, *70*(4), 603–623.

Berrone, S., Fidelibus, C., Pieraccini, S., & Scialò, S. (2014b). Simulation of the steady-state flow in discrete fracture networks with non-conforming meshes and extended finite elements. *Rock Mechanics and Rock Engineering*, *47*(6), 2171–2182.

Berrone, S., Pieraccini, S., & Scialò, S. (2013a). A PDE-constrained optimization formulation for discrete fracture network flows. *SIAM Journal on Scientific Computing*, *35*(2), B487–B510.

Berrone, S., Pieraccini, S., & Scialò, S. (2013b). On simulations of discrete fracture network flows with an optimization-based extended finite element method. *SIAM Journal on Scientific Computing*, *35*(2), A908–A935.
Berrone, S., Pieraccini, S., & Scialò, S. (2014a). An optimization approach for large scale simulations of discrete fracture network flows. *Journal of Computational Physics*, *256*, 838–853.
Berrone, S., Pieraccini, S., & Scialò, S. (2016). Towards effective flow simulations in realistic discrete fracture networks. *Journal of Computational Physics*, *310*, 181–201.
Berrone, S., Pieraccini, S., & Scialò, S. (2017). Non-stationary transport phenomena in networks of fractures: Effective simulations and stochastic analysis. *Computer Methods in Applied Mechanics and Engineering*, *315*, 1098–1112.
Berrone, S., Pieraccini, S., Scialò, S., & Vicini, F. (2015a). A parallel solver for large scale DFN flow simulations. *SIAM Journal on Scientific Computing*, *37*(3), C285–C306.
Bierig, C., & Chernov, A. (2015). Convergence analysis of Multilevel Monte Carlo variance estimators and application for random obstacle problems. *Numerische Mathematik*, *130*, 579–613.
Brenner, K., Hennicker, J., Masson, R., & Samier, P. (2016). Gradient discretization of hybrid-dimensional Darcy flow in fractured porous media with discontinuous pressures at matrix-fracture interfaces. *IMA Journal of Numerical Analysis*, *37*, 1551–1585.
Cliffe, K. A., Giles, M. B., Scheichl, R., & Teckentrup, A. L. (2011). Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, *14*, 3–15.
D'Angelo, C., & Scotti, A. (2012). A mixed finite element method for Darcy flow in fractured porous media with non-matching grids. *ESAIM: Mathematical Modelling and Numerical Analysis*, *46*(2), 465–489.
Dershowitz, W. S., & Fidelibus, C. (1999). Derivation of equivalent pipe networks analogues for three-dimensional discrete fracture networks by the boundary element method. *Water Resources Research*, *35*, 2685–2691.
Dick, J., Kuo, F. Y., & Sloan, I. H. (2013). High-dimensional integration: The quasi-Monte Carlo way. *Acta Numerica*, *22*, 133–288.
Faille, I., Fumagalli, A., Jaffré, J., & Roberts, J. E. (2016). Model reduction and discretization using hybrid finite volumes for flow in porous media containing faults. *Computational Geosciences*, *20*(2), 317–339.
Formaggia, L., Fumagalli, A., Scotti, A., & Ruffo, P. (2014). A reduced model for Darcy's problem in networks of fractures. *ESAIM: Mathematical Modelling and Numerical Analysis*, *48*, 1089–1116.
Giles, M. B. (2008). Multi-level Monte Carlo path simulation. *Operations Research*, *56*(3), 607–617.
Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328.
Graham, I., Kuo, F., Nuyens, D., Scheichl, R., & Sloan, I. (2011). Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications. *Journal of Computational Physics*, *230*(10), 3668–3694.
Huang, H., Long, T. A., Wan, J., & Brown, W. P. (2011). On the use of enriched finite element method to model subsurface features in porous media flow problems. *Computational Geosciences*, *15*(4), 721–736.
Huang, Z., Yan, X., & Yao, J. (2014). A two-phase flow simulation of discrete-fractured media using mimetic finite difference method. *Communications in Computational Physics*, *16*(3), 799–816.
Hyman, J., Gable, C., Painter, S., & Makedonska, N. (2014). Conforming Delaunay triangulation of stochastically generated three dimensional discrete fracture networks: A feature rejection algorithm for meshing strategy. *SIAM Journal on Scientific Computing*, *36*, A1871–A1894.
Hyman, J. D., Aldrich, G., Viswanathan, H., Makedonska, N., & Karra, S. (2016). Fracture size and transmissivity correlations: Implications for transport simulations in sparse three-dimensional discrete fracture networks following a truncated power law distribution of fracture size. *Water Resources Research*, *52*, 6472–6489. https://doi.org/10.1002/2016WR018806
Hyman, J. D., Karra, S., Makedonska, N., Gable, C. W., Painter, S. L., & Viswanathan, H. S. (2015). dfnWorks: A discrete fracture network framework for modeling subsurface flow and transport. *Computers & Geosciences*, *84*, 10–19.
Icardi, M., Boccardo, G., & Tempone, R. (2016). On the predictivity of pore-scale simulations: Estimating uncertainties with multilevel Monte Carlo. *Advances in Water Resources*, *95*, 46–60.
Karimi-Fard, M., Durlofsky, L., & Aziz, K. (2004). An efficient discrete-fracture model applicable for general-purpose reservoir simulators. *SPE Journal*, *9*, 227–236.
Kuo, F., Scheichl, R., Schwab, C., Sloan, I., & Ullmann, E. (2017). Multilevel quasi-Monte Carlo methods for lognormal diffusion problems. *Mathematics of Computation*, *86*, 1–32.
Le Maître, O. P., & Knio, O. M. (2010). Spectral methods for uncertainty quantification. In *Spectral methods for uncertainty quantification: With applications to computational fluid dynamics* (xvi+536 pp.). New York, NY: Springer.
Lenti, V., & Fidelibus, C. (2003). A BEM solution of steady-state flow problems in discrete fracture networks with minimization of core storage. *Computers & Geosciences*, *29*(9), 1183–1190.
Lu, D., Zhang, G., Webster, C., & Barbier, C. (2016). An improved Multilevel Monte Carlo method for estimating probability distribution functions in stochastic oil reservoir simulations. *Water Resources Research*, *52*, 9642–9660. https://doi.org/10.1002/2016WR019475
Makedonska, N., Painter, S. L., Bui, Q. M., Gable, C. W., & Karra, S. (2015). Particle tracking approach for transport in three-dimensional discrete fracture networks. *Computational Geosciences*, *19*(5), 1123–1137.
Martin, V., Jaffré, J., & Roberts, J. E. (2005). Modeling fractures and barriers as interfaces for flow in porous media. *SIAM Journal on Scientific Computing*, *26*(5), 1667–1691.
Mustapha, H., & Mustapha, K. (2007). A new approach to simulating flow in discrete fracture networks with an optimized mesh. *SIAM Journal on Scientific Computing*, *29*(4), 1439–1459.
Nobile, F., Tempone, R., & Webster, C. G. (2008). A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Scientific Computing*, *46*(5), 2309–2345.
Nœtinger, B. (2015). A quasi steady state method for solving transient Darcy flow in complex 3D fractured networks accounting for matrix to fracture flow. *Journal of Computational Physics*, *283*, 205–223.
Nœtinger, B., & Jarrige, N. (2012). A quasi steady state method for solving transient Darcy flow in complex 3D fractured networks. *Journal of Computational Physics*, *231*(1), 23–38.
Pichot, G., Erhel, J., & de Dreuzy, J. (2010). A mixed hybrid mortar method for solving flow in discrete fracture networks. *Applicable Analysis*, *89*, 1629–1643.
Pichot, G., Erhel, J., & de Dreuzy, J. (2012). A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks. *SIAM Journal on Scientific Computing*, *34*, B86–B105.
Pichot, G., Poirriez, B., Erhel, J., & de Dreuzy, J.-R. (2014). A Mortar BDD method for solving flow in stochastic discrete fracture networks. In Erhel, J., Gander, M., Halpern, L., Pichot, G., Sassi, T., & Widlund, O. (Eds.), *Domain decomposition methods in science and engineering XXI. Lecture Notes in Computational Science and Engineering* (Vol. 98, pp. 99–112). Cham: Springer.

Pieraccini, S., & Scialò, S. (2016). On a PDE-constrained optimization approach for flow simulations in fractured media. In *Advances in discretization methods. SEMA SIMAI Springer series* (Vol. 12, pp. 27–45). Switzerland: Springer International Publishing.

Sandve, T., Berre, I., & Nordbotten, J. (2012). An efficient multi-point flux approximation method for discrete fracture-matrix simulations. *Journal of Computational Physics*, *231*(9), 3784–3800.

Schwenck, N., Flemisch, B., Helmig, R., & Wohlmuth, B. (2015). Dimensionally reduced flow models in fractured porous media: Crossings and boundaries. *Computational Geosciences*, *19*(6), 1219–1230.

Svensk Kärnbränslehantering, A. B. (2010). *Data report for the safety assessment, SR-site* (Tech. Rep. TR-10–52). Stockholm, Sweden: SKB.

Tartakovsky, D. M., & Xiu, D. (2006). Stochastic analysis of transport in tubes with rough walls. *Journal of Computational Physics*, *217*(1), 248–259.

Teckentrup, A. L., Scheichl, R., Giles, M. B., & Ullmann, E. (2013). Further analysis of Multilevel Monte Carlo methods for elliptic PDEs with random coefficients. *Numerische Mathematik*, *125*, 569–600.

Xiu, D. (2010). *Numerical methods for stochastic computations*. Princeton, NJ: Princeton University Press.

Xiu, D., & Tartakovsky, D. M. (2006). Numerical methods for differential equations in random domains. *SIAM Journal on Scientific Computing*, *28*(3), 1167–1185.